

# **EXHIBIT 1**

# Third Expert Report: Assessment of MaverickMonitor Software Reliability

February 27, 2018

Prepared by Dr. Kal Toth, P.Eng, Portland, OR 97205

For Mr. J. Curtis Edmondson, Law Offices of J. Curtis Edmondson, Hillsboro, OR 97124

The purpose of this document is to report my assessment of the reliability of the MaverickMonitor also known as IPTRACKER (IPP International) and NARS (Excipio) (see [4d]). My reliability assessment is based on the evidence I have been provided to date. I have independently arrived at the opinions expressed in this report which depend on the accuracy of this evidence. My opinions are informed by my systems and software engineering qualifications, knowledge and experience.

## 1. Software Engineering Standards and Guidance on which I Rely

I rely on the following standards and guidance in support of my expressed opinions:

- a. *Software Engineering Institute's (SEI) Capability Maturity Model (CMM)*: has been refined and widely applied for over two decades assisting organizations choose and tailor the most appropriate software behaviors, practices, and processes in order to achieve software reliably and sustainably goals.
- b. *IEEE Software Engineering Standards including IEEE Std 12207, Systems and Software Engineering Software Life Cycle Processes*: common frameworks with well-defined terminology for developing software-based systems from the requirements stage to system retirement.
- c. *Validation of Forensic Tools and Software, A Quick Guide for the Digital Forensic Examiner* by Josh Brunty: relies on the *Daubert Standard* and NIST's Computer Forensic Tool Testing Project (CFTT) providing guidance for validating software-based systems - also discussed in my expert report [4e].

## 2. Most Relevant Qualifications

The opinions expressed in this report are drawn from my professional experience, detailed in the annex, where I highlight my most relevant qualifications for conducting this reliability assessment:

- a. Independent validation and verification (IV&V) for External Affairs Canada
- b. Quality, reliability, maintainability, safety, security, and software engineering for Hughes Aircraft
- c. Software engineering practice leader for CGI Group and Hughes Aircraft
- d. Software engineering courses for 10 universities including TechBC, Oregon State, and Portland State.

## 3. Evidence Reviewed and Referenced Herein

- a. Skype Deposition of Michael Patzer, October 13, 2016.
- b. Declaration of Michael Patzer, September 30, 2016.
- c. Expert Report, Patrick Paige, October 26, 2016.
- d. Supplemental Expert Report, Patrick Paige, December 16, 2016.
- e. Supplemental Report and Opposition to Kal Toth and Bradley Wittman's Expert Report, Michael Patzer, Dec. 30, 2016.
- f. Expert Report of Benjamin Perino, November 23, 2017.
- g. Functional Description, IPP International IPTRACKER v1.2.1 appearing as Exhibit 1 of Declaration of Tobias Fieser in Support of Plaintiffs Motion for Leave to Take Discovery ... filed 08/16/11.
- h. IPTRACKER software provided under Stipulated Protective Order Case No. 3:15-cv-00907-AC.
- i. Expert Witness of Dr. Simone Richter, April 2, 2014.
- j. Expert Report of Robert D. Young, February 11, 2015
- k. Deposition of Robert D. Young, January 2, 2018

#### 4. My Referenced Expert Reports

I have documented my reviews of some of the above documents in the following reports:

- a. *Expert Report Re. Malibu LLC vs. John Doe*, Kal Toth, Dec 14<sup>th</sup>, 2016. I pointed out the lack of evidence supporting Patzer's claim in [3a] that NARS is free of defects (is flawless) using the well-known Therac-25 case to illustrate. I also addressed the inadequacy of testing by Paige [3c].
- b. *Expert Report Re. Malibu LLC vs. John Doe, Rebuttal of Patzer Declaration and Paige Expert Report*, Dec. 28, 2017. I rebutted Patzer's declaration [3b] and Paige's supplemental expert report [3d] pointing out the absence of technical specifications, lack of software process, inadequate testing, etc.
- c. *Expert Report Re. Malibu LLC vs. John Doe, response to Patzer Supplemental Expert Report*, Kal Toth, Jan 6<sup>th</sup>, 2017. I rebutted several claims by Patzer [3e] including that an agile process was used.
- d. *Expert Report of Kal Toth Concerning Technical Report to Maverickeye*, May 10<sup>th</sup>, 2017. I compared the Maverickeye and Malibu technical reports demonstrating the equivalency of the systems used.
- e. *Second Expert Report of Kal Toth Regarding the Maverickeye Case*, Dec 24, 2017. I critiqued the "Functional Description" provided in the Declaration of Fieser [3g], and provided a preliminary analysis of the IPTRACKER source code [3h] observing that MaverickMonitor is adapted open source software.

#### 5. My Assessment of the Reliability of the Software

MaverickMonitor is a software-based system used for forensic purposes, namely, to detect alleged copyright infringement by Internet BitTorrent users.

##### 5.1 No Evidence MaverickMonitor Operates Reliably

Patzer asserts in [3a] that the system can detect the IP address of a defendant with 100% accuracy. He also states that other experts have claimed that the system works flawlessly. Perino in [3f] (para#16) claims "This infringement system accurately collected and recorded evidence proving that the IP addresses in this case infringed ...". Neither Patzer nor Perino provide evidence that the system is free of defects.

*Patzer, Perino and Richter have asserted that MaverickMonitor is 100% accurate and free of defects. In my experience, claiming that operational software is defect-free is not credible. Neither is asserting that a relatively complex software-based system like MaverickMonitor detects infringement flawlessly. If a system is expected to be highly accurate, and advertised as such, the software engineering and development process must be sufficiently capable and mature, and should be guided by credible standards such as the SEI CMM [1a], IEEE standards [1b] and NIST's CFTT guidance [1c].*

##### 5.2 No Evidence Capable and Mature Software Processes Guided Development

Professional software engineers establish capable software processes suitable for the requirements of each of their software projects. Plan-driven, agile, and hybrid software processes can be designed to meet project objectives. The central aim is to adopt and/or adapt a software process that enables the reliable construction of software-based systems that consistently meet functional, reliability, usage, performance and other requirements. Naturally, software-based systems that could compromise reputation, privacy, finances, and safety, should they fail, must meet an elevated standard. Although Patzer in [3e] claimed that his project team followed an "agile process", I point out in [4c] that his claim is without merit given his testimony in [3a] and [3b].

*No evidence has been provided that a capable software process [1a] was put in place to ensure that MaverickMonitor is operationally reliable.*

##### 5.3 No Evidence of a Theory of Operation or Technical Specification

In my experience, a well-articulated specification is necessary to guide the team when implementing a complex system like MaverickMonitor. It is critical that such a specification accurately state the intended purpose, unambiguously specifying the theory of how the planned system will operate to satisfy this purpose. The intended purpose is to implement a copyright infringement detection system that reliably detects infringing Bit

Torrent users under contemplated operational conditions. A *theory of operation* specification (a.k.a. operational concept) defines the technical details of how the software will correctly and consistently integrate with the Bit Torrent protocols to achieve the intended purpose. Development of the theory of operation specification requires the intimate involvement, scrutiny and agreement of stakeholders. Once ratified, additional specifications can be developed to guide the software development team.

*My reviews of Patzer, Fieser, Perino, Young and Richter, confirm that a well-articulated expression of the intended purpose of the system, including its theory of operation explaining how the system integrates with the Bit Torrent protocols, was not produced. Patzer [3a][3b][3e] confirms that he was not provided specifications, and that his team followed verbal instructions only. Fieser's declaration [3g] includes a "functional description" devoid of useful information as I explain in [4e]. Without the aid of formal theory of operation, Perino [3f], Young [3i], and Richter [3j] have asserted that the infringement system works correctly. Their opinions are offered without objective evidence (facts, data or analyses) proving that the system reliably satisfies the technical theory of how the system leverages the Bit Torrent protocols to consistently detect infringement. Without such technical specifications the system cannot be assumed to be detecting infringement reliably. The next two issues demonstrate the need for a theory of operation specification to identify a systemic problem.*

#### **5.4 Users Abandoning Bit Torrent Sessions Reported as Infringers**

I observed in [4c] that the infringement detection system reports infringement even if only a few pieces of a media file (e.g. movie) are detected. Richter [3j] takes the position that detecting even a sub-piece is enough to report infringement stating in 11.39, "... the intention of the user is to acquire a full copy of a movie", and in 11.42, "... possession and distribution of a sub-piece is a very strong indication that the user is or will be in possession of the complete data set."

This assertion prompted me to ask: *Is proof of possession of some pieces out of hundreds or thousands of pieces comprising a copyrighted media file, enough to demonstrate that a copyright has been infringed?*

I also observed that Richter fails to consider cases where users decide to abort a Bit Torrent session after launch because of confusion surrounding the actual title; or realizing that the content may be copyrighted; or for some other valid reason. However, when a user abandons his/her session, Bit Torrent protocols may already have shared a few pieces of content with members of the swarm. It is not possible for a member of a swarm to detect when other peers has abandoned the swarm.

*To demonstrate infringement, one should provide evidence not only that the alleged infringing user's Bit Torrent client software is in possession of the copyrighted media, but that the user is able to obtain useful benefit (e.g. entertainment) from the acquired media. Observe that if the user does not possess the entire media file (all pieces), the user's Bit Torrent client cannot create a readable copy, and the user will not be able to view the copyrighted media. Abandoning a Bit Torrent session before all pieces are downloaded would create such a circumstance. Nevertheless, MaverickMonitor reports such users as infringers.*

#### **5.5 MaverickMonitor Unable to Reliably Distinguish Infringers from Innocent Parties**

I also wondered why MaverickMonitor routinely reports infringement when only a few pieces are detected, and why the system seems to rarely download all the pieces of targeted media files. Given a technical specification such as a theory of operation had not been produced, I looked more closely at the plaintiff's expert reports and depositions. Young [3j] confirms that as a Bit Torrent swarm ramps up, the Bit Torrent clients of interested users including MaverickMonitor, coordinated by a Bit Torrent tracker, begin to send requests for pieces. Participants, including MaverickMonitor, receive different pieces for the same media file from many members of the swarm. MaverickMonitor receives pieces of the media file from peer users in the swarm, but only receives a portion of the pieces from any given peer user. Once the MaverickMonitor system has received all pieces of the media file, peer users stop sending pieces to the system which reports such peers as infringers even though only a subset of the pieces were detected.

To validate my analysis, I first examined Richter [3j] 11.37 who confirms "... the System does not acquire the whole file from a single source. This is not necessary or always possible." Her comment implies that MaverickMonitor may only be capable of reliably determining that a user has downloaded all pieces of a media file in special cases, for example, when only a single user is interested in acquiring a given media file seeded by MaverickMonitor. This explains why Paige and Richter conducted rudimentary confidence tests configuring only a handful of Bit Torrent clients and test files (see 5.11 for more).

*I conclude that MaverickMonitor can only download all the pieces of a targeted file in special cases such as the simple confidence tests conducted by Paige and Richter, or when peers in the Bit Torrent community are either not interested in or unaware of the existence of media alleged to be infringed. In most cases, the system can only download a subset of the pieces from any given user. This means that MaverickMonitor is not able to reliably distinguish between actual copyright infringers whose pieces are only partially detected, and those users who have aborted their sessions distributing only a small number of pieces detected by MaverickMonitor. MaverickMonitor is not able to reliably distinguish between infringers and innocent users.*

## **5.6 No Evidence Patches and Updates have been Applied to the Open Source Software**

The source code of MaverickMonitor, claimed to be proprietary, was made available to me under a protective order [3h]. It appears that most of the source code is drawn from two open source software systems, namely, MonoTorrent version 1.1.16.1+, and SharpPcap (no version number was provided).

As documented in my second expert report [4e] regarding MaverickEye, the subsystems composing these two open source code bases contain notices by authors declaring their copyrights (2003 and 2013) to MonoTorrent and SharpPcap software modules. Authors McGovern, Dufour, Maurier, Laval, Pozdnyakov, Buguer, Zaroni, Burger, Bockover, Torstensson, Coleman, and Javier appear to be members of the open source community. The only software license notice in the code was asserted by McGovern and Dufour. The SharpPcap code base specifies it should be compiled with the MonoTorrent code base.

I also inspected the code to locate custom modules designed to implement the intended purpose, namely, to create an operational "infringement detection system" as explained by Perino [3f] (para#11). I searched the code for copyright notices and software licenses attributable to non-open source entities such as Perino, Guardaley, IPP International, and Excipio. I did not find evidence of any such notices. It is therefore not possible to identify the custom (proprietary) modules or assess whether these modules were properly inspected, tested, and debugged prior to being released for operational use by MaverickEye.

I observed that a notice (TODO.txt) embedded in the code by one of the lead developers declared that the MonoTorrent software contains many bugs ("I am sure there are many").

The MonoTorrent and SharpPcap development sites list numerous open and closed bug reports. There is no evidence that any of the patches and updates to repair these bugs have been applied to the software.

*The reliability of MaverickMonitor cannot be assessed without objective evidence that recommended patches and updates released by the open source development teams, namely, MonoTorrent and SharpPcap, have been applied or installed. Such evidence has not been provided.*

## **5.7 MaverickMonitor Likely Contains Many Undetected Defects**

My review of the software provided under protective order [see 3b.] reveals that the MaverickMonitor software is comprised of over 140,000 lines of code (LOC) consuming about 6.6M bytes of storage. In my experience this represents a large code base.

Patzer simply states in [3a] that "lots of tests were run". Given the lack of objective evidence, I conclude that testing was ad hoc and light weight. Under conditions of light-weight testing, a useful rule of thumb is that the software contains 6-7 latent defects (bugs) per 1000 lines of code. This means that a software engineer or software tester should expect the software to contain between 800 to 1000 latent defects, and should plan, accordingly, to conduct tests that remove at least the critical and major defects.

*MaverickMonitor has a large code base of over 140,000 lines of source code (LOC). This means that the number of latent (undetected) defects could be quite large. There is no evidence of effective testing.*

## **5.8 MaverickMonitor Software is Computationally Complex**

Functional complexity and software size (140,000 LOC) are strong indicators of the complexity of the implementing code. When activated by the operator, MaverickMonitor scans and simultaneously tracks tens of thousands of BitTorrent users in a so-called “swarm” of users sharing digital media (e.g. software updates and movies). Each participating user’s personal computer splits up each shared file into thousands of pieces; calculates numerical hashes for each piece; shares these pieces with thousands of other users; each recipient’s computer receives thousands of pieces, each typically from different users; and each user’s computer verifies the numerical hashes of thousands of pieces, assembles them into coherent files, and saves them on the recipient computer’s hard drive. The MaverickMonitor system must be capable of reliably collecting thousands of such pieces in real-time, simultaneously detecting the correct IP address of every received piece, binding each of the pieces to the sender’s IP address, sorting all of the pieces by IP address, simultaneously storing all the pieces for every IP address into a database, also in real time, and generating reports that correctly associate each IP address with the pieces of each file received from the IP addresses of all senders.

*Maverickeye is computationally complex given the software performs a large number of complex tasks simultaneously. From a development perspective, the number of opportunities for programmers and system integrators to make mistakes and thereby introduce defects (“bugs”) into the code is very high. In my experience, software of this computational complexity must undergo a rigorous software engineering process to ensure that critical and major defects are removed before deployment.*

## **5.9 Architectural Design Specifications were Not Developed**

System architecture and software design control system behavior. Typically, architectural design is expressed in the form of block diagrams used to represent software subsystems, components, and modules; activity diagrams depicting software logic; sequence and collaboration diagrams showing timing and sequencing among processes; and state diagram/charts modeling modes of operation. Such representations are essential for the development of complex systems needing to reliably control timing and synchronization among processes, inputs, outputs, and events under various operating conditions.

*Patzer in [3a] confirmed that his team did not receive technical specifications or develop block diagrams or flowcharts to guide his team’s development effort. Young [3k] created only a crude block diagram purporting to represent the MaverickMonitor system. Without architectural design specifications it is not possible to test and thereby verify that components, subsystems, and the integrated system are operating correctly and reliably.*

## **5.10 No Evidence of Reviews, Walkthroughs and/or Inspections**

Static analysis of technical specifications in the form of reviews, walkthroughs and code inspections can detect and prevent logic, process synchronization, and timing errors, significantly increasing software reliability. Deep system timing, concurrency, and integrity problems cannot be detected by simple testing.

*Richter in [3i] (sections 7.2 and 7.3) describes how she analyzed the code, line by line, to detect logic and other errors. Given the code base consists of over 140,000 lines of code, I venture that it would have been infeasible for her to review all the code. Furthermore, the lack of technical specifications would have made it impossible for her to determine if the code correctly implements functionality as intended. Other than her assertions, there is no evidence that static analysis of any kind was conducted during development.*

## **5.11 No Evidence of Adequate Component, Subsystem, Integration and System Testing**

Experienced software engineering professionals know that software, during both development and maintenance, is bound to contain latent defects, some of which could have major impact on operational capability if not persistently ferreted out. Systematic testing is therefore essential to verify that a system meets



functional requirements and satisfies technical specifications. Testing should be performed at multiple integration levels to remove software bugs when the system is complex and is intended to operate reliably. Effective testing involves specifying and executing test cases that include test descriptions, test criteria, test procedures, test data and expected test results. Periodic regression testing is needed to ensure continuous reliability assurance (see 5.10 below).

*No evidence has been provided that adequate testing was conducted during development. Patzer in [3a] confirmed that his team did not test the bit torrent software to verify that it operated correctly. Paige [3c] [3d] conducted a simple confidence test asserting his tests confirm that the “infringement detection system works” - not stating how consistently or reliably. Richter [3i] (section 9) conducted similar rudimentary tests. Neither Paige nor Richter configured or ran tests simulating the actual operating environment, namely, a swarm of Bit Torrent computers at many unknown locations sharing a significant number media files among peer computers in the swarm, a tracker, and the MaverickMonitor system.*

## **5.12 No Evidence System was Regularly Validated**

Because MaverickMonitor is designed to identify and track potential copyright infringers, one can assume that accepted methods for validating forensic tools and software should apply to provide assurances that the software is operating correctly and accurately. In my second MaverickEye expert report [4e], I summarized the observations and insights of Josh Bunty (Marshall University Forensic Science Center) in his paper entitled “Validation of Forensic Tools and Software: A Quick Guide for the Forensic Examiner”. His paper is informed by the work of the National Institute of Standards and Technology (NIST).

*With respect to forensic tools and software, Bunty asserts that the Daubert standard requires an independent judicial assessment of the reliability of the scientific test or method used. Bunty describes four basic steps for planning and testing forensic tools, and explains that software-based tools used for forensic purposes should be validated quarterly to achieve repeatable and reproducible results. No evidence has been provided that MaverickMonitor is validated quarterly or on any periodic schedule.*

## **5.13 No Evidence of a Problem Tracking and Resolution**

Reviews, walkthroughs, inspections, and testing processes identify problems in technical specifications, the code, and the runtime system. Good software engineering practice for proprietary and open source development and maintenance employs problem tracking and resolution tools together with systematic procedures to track bugs and issues through to resolution, prioritizing critical and major defects.

*There is no evidence that capable problem tracking procedures and tools were used to track software bugs and fixes through to resolution throughout development as well as in production.*

## **5.14 No Evidence Configuration Management Used to Control Software Releases**

To ensure system integrity, it is critical for operational software-based systems to control the revisions of their evolving software configurations from release to release throughout both development and operational lifecycles. Young in [3k] disclosed that he did not know whether the software version of MaverickMonitor running at the time of the alleged infringement was compiled from the same version of source code as inspected by Richter [3i]. Young examined version 1.5.1 while it appears that I received version 1.1.16.1+ of the software under the protective order. Furthermore, the source code provided to me contains no release notes indicating whether the code corresponds with the runtime version at the time of the alleged infringement.

*There is no evidence that the MaverickMonitor software configuration has been systematically managed to control releases (versions) of the software to protect the integrity of the system from release to release.*

## **5.15 No Evidence of Software Quality Assurance or Independent Validation and Verification**

Professional software engineers establish a separate quality assurance team to conduct routine internal reviews and audits of the software process including plans and procedures, technical specifications, records of reviews,

inspections, and tests, and reports provided to customers. For complex projects, contractors are sometimes commissioned to independently validate and verify (IV&V) software development.

*There is no evidence that either quality assurance or independent verification and validation were conducted to assure system reliability.*

## **5.16 No Evidence Critical Failure Modes Were Analyzed**

The system's essential integrity constraint is that numerous fragments of copyrighted content from many sites must be consistently and correctly bound to the correct IP address, and that this binding of information must be correctly and consistently written into the data repository. Non-obvious system failure modes under normal and abnormal operating conditions could corrupt this essential integrity constraint.

*There is no evidence that the risks associated with critical failure modes were analyzed or mitigated to ensure that the reliability of the MaverickMonitor production system is not compromised during operational use. Perino, Richter, and Young do not consider or discuss the risks associated with timing and synchronization among concurrent tasks, input streams and database transactions. The possibility of race conditions, deadlocks and aborted software processes could corrupt software logic designed to bind the IP addresses of Bit Torrent users with downloaded fragments (pieces) of shared media (files, movies, etc).*

## **6. Summary of Findings and Recommendations**

- a. The plaintiff's experts have asserted that MaverickMonitor is 100% accurate and free of defects. In my experience, claiming that operational software is defect-free is not credible. Neither is asserting that a complex software-based system detects infringement flawlessly. If a system is expected to be highly accurate, and advertised as such, the software engineering and development processes must be sufficiently capable and mature, should be guided by credible standards, and be supported by experienced personnel and proven tools. There is no evidence that capable software processes, technical specifications, architectural designs, problem tracking and resolution, configuration release management, and quality assurances were put in place to ensure the system detects infringement correctly, consistently and reliably.
- b. My reviews of Patzer, Fieser, Perino, Young and Richter confirm that a well-articulated expression of the intended purpose of the system, including its theory of operation explaining how the system integrates with the Bit Torrent protocols, was not produced. Patzer confirms that he was not provided specifications, and that his team followed verbal instructions only. Fieser's declaration includes a "functional description" devoid of useful information. Without the aid of formal theory of operation, Perino, Young, and Richter have asserted that the infringement system works correctly. Their opinions are offered without facts, data or analyses proving that the system reliably satisfies the technical theory of how it leverages the Bit Torrent protocols to reliably detect infringement. The next two issues demonstrate this problem.
- c. Evidence of infringement should show that the alleged infringing user's Bit Torrent software is in possession of the copyrighted media, and that the user is able to obtain useful benefit (e.g. entertainment) from the acquired media. Observe that if the user does not possess the entire media file (all pieces), the user's Bit Torrent client cannot create a readable copy, that is, the user will not be able to view the copyrighted media. Abandoning a Bit Torrent session before all pieces are downloaded would create such a circumstance. Nevertheless, MaverickMonitor reports users who have aborted their sessions as if they were infringing.
- d. I conclude that MaverickMonitor can only download all the pieces of a targeted file in special cases such as the simple confidence tests conducted by Paige and Richter, or when peers in the Bit Torrent community are either not interested in or unaware of the existence of media alleged to be infringed. In most cases, the system can only download a subset of the pieces from any given user. This means that MaverickMonitor is not able to reliably distinguish between actual copyright infringers whose pieces are only partially detected, and those users who have aborted their sessions distributing only a small number of pieces detected by the system. Hence, MaverickMonitor is not able to reliably distinguish between infringers and innocent users.
- e. The reliability of MaverickMonitor cannot be assessed without objective evidence that recommended patches and updates released by the open source development teams, namely, MonoTorrent and SharpPcap, have been applied or installed. Such evidence has not been provided.
- f. MaverickMonitor has a large code base of over 140,000 lines of source code (LOC). This means that the number of latent (undetected) defects could be quite large. Maverickeye is also computationally complex

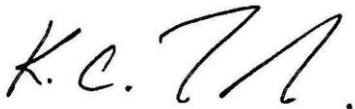


given the software performs a large number of complex tasks simultaneously. The number of opportunities for programmers and system integrators to make mistakes and thereby introduce defects (“bugs”) into the code is very high. A rigorous software engineering process should have been conducted to ensure critical and major defects were removed before deployment. No evidence of such processes was provided.

- g. Richter describes how she analyzed the code, line by line, to detect logic and other errors – she found none. Given the large code base of over 140,000 lines of code, I venture that it would have been infeasible for her to review all the code. Furthermore, the lack of technical specifications would have made it impossible for her to determine if the code correctly implements the required functionality as intended.
- h. No evidence provided that adequate testing was conducted during development. Patzer confirmed that his team did not test the Bit Torrent software to verify correctness. Paige conducted a simple confidence test asserting the “infringement detection system works”. Richter conducted similar rudimentary tests. Neither expert configured or ran tests simulating the actual operating environment of many Bit Torrent peers at unknown locations sharing numerous media with each other including the MaverickMonitor system.
- i. There is no evidence that capable problem tracking procedures and tools were used to track software bugs and fixes through to resolution throughout development as well as in production. Furthermore, there is no evidence that the MaverickMonitor software configuration has been systematically managed to control releases (versions) of the software to protect the integrity of the system from release to release.
- j. MaverickMonitor is a forensic tool used to detect copyright infringement. In [4e] I describe Josh Bundy’s assertion that the Daubert standard requires an independent judicial assessment of the reliability of such a tool by validating quarterly to ensure repeatable and reproducible results. No such evidence was provided.
- k. There is no evidence that the risks associated with critical failure modes were analyzed or mitigated to ensure that the reliability of the MaverickMonitor production system is not compromised in production.

In the absence of verifiable evidence, an objective software professional cannot conclude that MaverickMonitor detects the IP addresses of infringing bit torrent users correctly, consistently and reliably.

My rate is \$350.00 per hour.



Signed under the Penalty of Perjury,  
Kal Toth (Kalman C. Toth), Ph.D., P.Eng.

## Annex: My Most Relevant Experience and CV (Kal Toth)

---

## **My Most Relevant Qualifications**

My name is Kal Toth. I have a Ph.D. in computer engineering from Carleton University and am a professional engineer (P.Eng.) with a software engineering designation registered in BC.

I have practiced in the fields of software and quality engineering, information security, e-commerce, mobile systems, and distributed database systems. My detailed CV below covers my work history and key projects in industry and at universities, also listing my conference and journal publications, industry reports, university courses, and delivered seminars.

## **Independent Validation and Verification (IV&V)**

As Vice President of Systems Engineering for the CGI Group, I led a 3<sup>rd</sup> Party Validation and Verification team hired by the Canadian Federal Government to oversee their prime contractor's \$50M development of a security-critical global messaging system for Canada's embassies abroad. The primary purpose of this project was to ensure that the prime contractor's development teams developed adequate plans, requirements specifications, designs and test procedures, and executed their plans, reviews and procedures according to their obligations and standards called up under their contract with the Canadian government.

## **Quality, Reliability, Maintainability, Safety, Security, and Software Engineering**

As Director of Quality at Hughes Aircraft of Canada, Systems Division, I lead my team's quality assurance, reliability, maintainability, availability and safety engineering tasks supporting the development of five (5) large software-intensive Air Traffic Control (ATC) systems, including Canada's new ATC system, a \$500M project. I also supported the security working group for the project. My responsibilities included leading the development of our division's new software development methodology, including software requirements, architecture, development, testing and metrics processes, promulgating the division's transition from a traditional plan-based software process to a more flexible iterative software development process.

## **Software Engineering Practice Leader**

At CGI Group (VP Total Quality) and Hughes Aircraft (Director of Quality) I also had the role of software and systems engineering practice leader. I organized working groups and gave seminars aimed at developing skills in the areas of software project management, software processes, software quality assurance, professional issues, process improvement, and metrics.

## **Software Engineering Programs and Courses**

I later joined academia as an Associate Professor teaching software engineering, architectural design, quality, and project management courses to working professionals at the Technical University of BC, the University of British Columbia, Simon Fraser University, Oregon State University and Portland State University. I was the Director of the Oregon Master of Software Engineering program and the Executive Director of the WestMost consortium teaching software technology courses to working professionals across nine (9) universities in western Canada.

## **VP Engineering for a Real-Time Web-Centric Real-Time Alert System**

As Vice President of Engineering for Datalink Systems Corp I managed an agile team of ten software engineers, programmers and testers developing and maintaining a real-time alert system sending stock quotations to mobile devices of customers managing their portfolios online. The system ran on a server farm of a dozen physical servers, supported by an SQL database system. I established and shaped an iterative software development process for the team including functional and design specification, peer-reviews, independent module testing and system integration testing.

**Kalman C. Toth** Ph.D., P. Eng.304-1132 SW 19<sup>th</sup> Ave Portland OR 97205  
kalmancto@gmail.com 503.984.3531***Security, Software, Quality, and Systems Engineering Professional*****Background / Experience:**

- In leadership positions with technology companies in the fields of security, software and IT
- Software, systems and security-related engineering innovator, consultant, and change agent
- Technology solutions and consulting in government, financial and selected industry sectors
- Cybersecurity, identity management, e-commerce, mobile computing, distributed systems, networking, communications, and databases.
- Air traffic control; real-time stock quotation for mobile devices; search and rescue system, security devices and gateways; global secure messaging network; on-line learning systems
- Systems engineering evangelist: traditional and agile software development, project management
- Software engineering, IT and project management courses and training for working professionals.

**Competencies:**

Systems, security, software and quality engineering, Strategic and business planning, Project management, Digital Identity technology and security engineering, e-learning/distance education

**Citizenship and Residency:** U.S. Citizen, U.S. Resident, also a Canadian Citizen

**Languages:** English (mother tongue), Hungarian (father tongue), and French fluency

**World:** Early IT career with World Health Organization, Geneva, Switz; well-travelled in Europe

**Education:**

B. Eng. Electrical Engineering

M. Eng. Systems Engineering and Computer Science

Ph.D. Computer Systems Engineering

**Professional Engineer (P. Eng.):** BC Association of Professional Engineers and Geoscientists

**Training/Education Courses:** E-commerce, SW engineering, project management, prof. issues (i.e. IP)

**Pacific Northwest Software Quality Conference:** Board member and 2013 Conference Chair

**Portland State University:** Faculty Senate Budget Committee; Intellectual Property/DistEd Taskforce

**Goose Hollow, Portland Oregon:** neighbourhood association Board of Directors

**Patent:** "Electronic Identity and Credentialing System", US Patent No. 9646150, Apr 20/17.

**Patent:** "Methods for Using Digital Seals for Non-Repudiation of Attestations", Aug 20/17

**Patent-Pending:** "Registering and Acquiring E-credentials using Proof-of-Existence & Digital Seals", Feb 18, 2018, No. 15/898,217.

**Patent-Pending:** "Portable Caching System" submitted in 2007, abandoned in 2015

**Expert Reports:** copyright infringement cases, multiple expert reports, depositions

**Key Positions / Appointments:** listed

**Expert Reports:** listed

**Publications, Industry Reports, and Courses:** listed

See also <http://www.linkedin.com/pub/kal-toth/2/60b/b19>

## **Key Positions / Appointments**

### **NexGenID (2013 - 2014), CEO and CTO**

- Created innovative identity and credentialing technology: “Electronic Identity and Credentialing Technology” per above-referenced patent and patent-pending identity technology
- Developed detailed functional specification and proof-of-concept for digital identity prototype\_(Android-based)

### **aTrust Inc. (2012 - 2013), Chief Technology Officer (CTO)**

- Progressed startup’s vision for digital identity, technology roadmap, and product-line development strategy
- Built and maintained partner/vendor relationships in technology and banking sectors
- Managed and evaluated the distributed development team’s progress and performance

### **Portland State University (2003-12), Executive Director and Associate Professor**

- Directed, enhanced and evolved the Oregon Master of Software Engineering (OMSE) into a fully online learning program for working software professionals in Oregon’s hi-tech sector
- Delivered software engineering, project management, quality engineering, distributed team, estimating, and architectural design courses and seminars – both face-to-face and online
- Investigated identity management technologies targeted at the healthcare and banking sectors creating the “Persona Concept”, a framework for managing electronic credentials and private data of users across PCs, smart cards, smart phones, and other personal devices

### **Oregon State University (2001-2003), Associate Professor Computer Science**

### **Technical University of British Columbia (1999-2001), Assoc. Professor Information Technology**

### **Datalink Systems Corp. (1997-99), Vice President Engineering**

- Following a light-weight agile software development process, directed development and operations
- Led the development of a web-based service and payment processor for delivering real-time stock quotes, news, sports, and other services to wireless devices - pagers and cell phones
- Worked with marketing/support to develop requirements and rapid response to user problems
- Removed security weaknesses of the previously deployed service center
- Developed replacement architecture with scalability, backup and recovery features

### **Hughes Aircraft Systems Division (1992-95), Director of Quality**

- Led quality, reliability, maintainability, availability and system safety teams for five (5) large air traffic control projects (Canada, Canadian military, Switzerland, Indonesia and China)
- Leading member of the core team transitioning division from a waterfall to an iterative software process which guided the development of Canada’s \$400M air traffic control system (“CAATS”)
- Created a new process infrastructure for the division’s policies, practices and procedures

### **CGI Group Inc. (1988-1992), Vice President Systems Engineering, Vice President Total Quality**

- Practice leader across CGI’s 10 regional offices for project management, software engineering, quality engineering, configuration management, and software estimating
- Led process improvement initiatives across CGI’s US and Canadian offices
- Developed and initiated a strategic plan to implement a company-wide total quality process
- Conducted independent verification and validation of a \$50M project to develop a globally secure network across Canada’s embassies abroad for External Affairs Canada
- Developed an innovative information security analysis model for Defence Canada

### **Intellitech Canada Ltd. (1983-88), Founder and President**

- Founded Intellitech, growing it into a 25-person systems engineering and consulting firm
- Conducted numerous design and development projects for distributed information systems, networks and security gateways for military, government and industry clients
- Led the development of Intellitech’s secure packet-network product and the delivery of prototypes to Communications Canada – funded by the Canadian National Research Council and the Bank of Montreal, and sponsored by the Communications Security Establishment

### **Carleton University (1980-83), Assistant Professor, Systems Engineering and Computer Science**



**Expert Reports: Intellectual Property (copyright infringement) Cases**

- Expert reports (3) for JC Edmondson law office for defendant in a copyright infringement case, 2016-17
- Expert reports (3) for JC Edmondson another defendant in a copyright infringement case, 2017-2018

**Publications and Seminars in the Field of Security, Identity and Authentication**

- Kalman C Toth, Brewing Next Generation Identity, Pacific Northwest Software Quality Conference, Oct 2015
- Kalman C. Toth, A Practical Identity Management Reference Implementation, International Conference on Computers and Their Application (CATA), Honolulu, Hawaii, March 28-30, 2007
- Kalman Toth, Persona Concept for Web-Based Identity Management, 2006 International Conference on Privacy, Security and Trust, UOIT, Newmarket, Ontario, Oct 30-November 1 2006
- "Identity Management Systems", tutorial for IEEE International Computer Software and Applications Conference (COMPSAC), Chicago, September 2006
- Information security seminars for the Assoc. of Prof. Engineers and Geoscientists of B.C., 2002 and 2006
- K.C. Toth, M.Subramaniam, Requirements for the Persona Concept, Requirements for High Assurance Systems (RHAS'03) workshop, Monterey, CA, September 9, 2003
- K.C. Toth, M. Subramaniam, The Persona Concept: A Consumer-Centered Identity Model, MobEA (Emerging Applications for Wireless and Mobile Access), Budapest, Hungary, May 2003
- K.C. Toth, M. Subramaniam, Persona Concept for Privacy and Authentication, International Business & Economics Research Journal, June 2003
- K.C. Toth, M. Subramaniam, I. Chen, Persona Concept for Privacy and Authentication, International Applied Business Research Conference, Acapulco, Mexico, March 2003; recipient of best paper award
- K.C. Toth, M.Donat and J. Joyce, Generating Test Cases from Formal Specifications, 1996 International Council of Systems Engineering (INCOSE) Symposium, July 1996
- M.W.L. Dennison, K.C. Toth & J.F. Clayton, Using a Practical Approach to Threat/Risk Analysis, Third Annual Canadian Computer Security Conference, Ottawa, May 14-16, 1991
- K. Toth, Information Security Architectures, AFCEA '90 (Armed Forces Communications & Electronics Association Technical Conference), Hawaii, November, 1990
- K.C. Toth, Security Architectures for Information Networks, AFCEA Canada '90, April 1990
- H. Adra, J. Allen, K. Toth, Trusted Integrated Project Support Environments, Second Annual Canadian Computer Security Conference, Ottawa, March 1990
- K. Toth, Towards an Improved Information Security Model, 1st Canadian Comp. Security Conf, January 1989
- K. Toth, Security Management in Data Networks, 1st Annual Canadian Computer Security Conf, Jan 1989
- AC Capel, C Laferriere & K.C Toth, Protecting the Security of X.25 Comm's, Data Com Mag, November 1988
- M.W.L. Dennison, K.C. Toth & J.F. Clayton, Using a Practical Approach to Threat/Risk Analysis, Third Annual Canadian Computer Security Conference, Ottawa, May 14-16, 1991
- K. Toth, Information Security Architectures, AFCEA '90 (Armed Forces Communications & Electronics Association Technical Conference), Hawaii, November, 1990
- K.C. Toth, Security Architectures for Information Networks, AFCEA Canada '90, April 1990
- K. Toth, Towards an Improved Information Security Model, 1st Canadian Comp. Security Conf, January 1989
- K. Toth, Security Management in Data Networks, 1st Annual Canadian Computer Security Conf, Jan 1989
- AC Capel, C Laferriere & K.C Toth, Protecting the Security of X.25 Comm's, Data Com Mag, November 1988
- System Security and Recovery Procedures, Datalink Systems Corp, January 1999
- "EDI and Security", CGI Group report, Dec. 1990
- "COSICS Security Verification Plan", Intellitech report to External Affairs, December, 1988
- "Information Security Model", report to National Defence, November 15, 1988
- "Data Encryption Equipment Specification", Internal report specifying the components of CryptoNet, Intellitech's X.25/DES product, 1986
- "A New Implementation Strategy for Secure Operating Systems", Intellitech Report, March 1986
- "Design and Security Considerations for a Gateway to Interconnect SAMSON and DATAPAC", Report to the Department of National Defence, 1980

## Conferences and Journal Publications

- Kalman C Toth, Brewing Next Generation Identity, Pacific Northwest Software Quality Conference, Oct'15
- Kalman C Toth, Herm Migliore, Critical Factors Characterizing Projects & Lifecycle Models, PNSQC, Oct'13
- Kalman Toth, Learning Software Engineering Online, Pacific Northwest Software Quality Conference, Oct'11
- Kal Toth, Organizational Approach for Sustaining E-Learning in Large Urban University, Future of Ed, Jun'11
- Kal Toth, Software Engineering Online and Hybrid Learning Models at PSU, CATA, March, 2011
- Kal Toth, Raleigh Ledet, Lessons Learned about Distributed Software Team Collaboration, PNSQC, Oct'10
- Kal Toth, Software Estimating: Navigating to Landing Zone, Computers & their App's, Honolulu, HI, Mar'10
- Kal Toth et. al., Distributed Software Engineering Team Collaboration, poster session, PNSQC, October 2009
- Kal Toth, Software Estimating, Flexibility and Principled Negotiation, Computers and their Applications in Industry and Engineering (CAINE), San Francisco, November, 2009
- Kal Toth, Selecting Software Estimating Techniques that Fit the Software Process, Pacific Northwest Software Quality Conference (PNSQC), Portland, Oregon, October, 2008
- Dan Brook, Kal Toth, Levels of Process Ceremony for Software Configuration Management, Pacific Northwest Software Quality Conference (PNSQC), Portland, Oregon, October, 2007
- Kalman C. Toth, A Practical Identity Management Reference Implementation, International Conference on Computers and Their Application (CATA), Honolulu, Hawaii, March 28-30, 2007
- Kal Toth, Experiences with Open Source Software Engineering Tools, IEEE Software, Nov/Dec 2006
- Kalman Toth, Persona Concept for Web-Based Identity Management, 2006 International Conference on Privacy, Security and Trust, UOIT, Newmarket, Ontario, Oct 30-November 1 2006
- L. Grove, R. Hickman, W. Matthews, K. Toth, Open Source Software Engineering Tools, Pacific Northwest Software Quality Conference (PNSQC), Portland, Oregon, October 12-13, 2004
- K.C. Toth, M.Subramaniam, Requirements for the Persona Concept, Requirements for High Assurance Systems (RHAS'03) workshop, Monterey, CA, September 9, 2003
- K.C. Toth, M. Subramaniam, The Persona Concept: A Consumer-Centered Identity Model, MobEA (Emerging Applications for Wireless and Mobile Access), Budapest, Hungary, May 2003
- K.C. Toth, M. Subramaniam, Persona Concept for Privacy and Authentication, International Business & Economics Research Journal, June 2003
- K.C. Toth, M. Subramaniam, I. Chen, Persona Concept for Privacy and Authentication, International Applied Business Research Conference, Acapulco, Mexico, March 2003; recipient of best paper award
- K.C. Toth and S. Nagboth, A Constraint-Based Personalization Model for E-Business Applications, International Applied Business Research Conference, Acapulco, Mexico, March 2003
- K.C. Toth, S. Nagboth, Intelligent Agents for Business Applications Using Constraint-Based Personalization, International Business & Economics Research (IBER) Journal, May 2002
- K.C. Toth, Software Product Evolution in the Classroom, American Society for Engineering Education / PSW, Fresno, California, April 8, 2002
- K.C. Toth, Simulating (Software) Product Evolution in the Classroom, The Western Canadian Conference on Computing Education (WCCCE), Nelson, British Columbia, May 3, 2001
- K.C. Toth and H. Todino, Instant Internet Intelligence for Wireless Business Applications, International Applied Business Research Conference, Cancun, Mexico, March 2001
- D Cyr, H Trevor-Smith, T Schiphorst & K.C Toth, A Web-Enabled Case Study in Project Management, International Business Education and Technology Conference, Cancun Mexico, March 2001
- K.C. Toth, M.Donat and J. Joyce, Generating Test Cases from Formal Specifications, 1996 International Council of Systems Engineering (INCOSE) Symposium, July 1996
- R. John, J. Madhur, R. Stewart, K. Toth, Software Quality Metrics Process For Large Scale Systems Development, 1996 INCOSE Symposium, July 1996
- K.C. Toth, J.J. Joyce, J. Masters, G. Pelletier, Precise, Unambiguous, Machine-Readable ATC Standards: Use of "Formal Methods" in the ATC Industry, ATCA Conference Proceedings, September 1995
- K Toth & J. Joyce, Industrialization of Formal Methods Through Process Definition, feature paper at the 1995 National Council on Systems Engineering Symposium, July 1995
- T. Paine, P. Kruchten & K. Toth, Modernizing ATC Through Modern Software Methods, Proceedings of the 38th Annual Air Traffic Control Association, Nashville, Tennessee, October 1993
- M.W.L. Dennison, K.C. Toth & J.F. Clayton, Using a Practical Approach to Threat/Risk Analysis, Third Annual

Canadian Computer Security Conference, Ottawa, May 14-16, 1991

- K. Toth, Information Security Architectures, AFCEA '90 (Armed Forces Communications & Electronics Association Technical Conference), Hawaii, November, 1990
- K.C. Toth, Security Architectures for Information Networks, AFCEA Canada '90, April 1990
- H. Adra, J. Allen, K. Toth, Trusted Integrated Project Support Environments, Second Annual Canadian Computer Security Conference, Ottawa, March 1990
- K. Toth, Towards an Improved Information Security Model, 1st Canadian Comp. Security Conf, Jan 1989
- K. Toth, Security Management in Data Networks, 1st Annual Canadian Computer Security Conf, Jan 1989
- AC Capel, C Laferriere & K.C Toth, Protecting the Security of X.25 Comm's, Data Com Mag, November 1988
- K.C. Toth, S.A. Mahmoud, J.S. Riordon, Query Processing Strategies in a Distributed Database Architecture, Distributed Data Systems, North-Holland Publishing Co., 1982
- K.C. Toth, S.A. Mahmoud & J.S. Riordon, An Approach to Query Processing in Distributed Databases, Proceedings of the Sixth International Conference on Very Large Data Bases, Montreal, 1980
- Kalman C. Toth, Distributed Database Architecture & Query Processing Strategies, Ph.D. Carleton U 1980
- S.A. Mahmoud, J.S. Riordon & K.C. Toth, Distributed Database Partitioning & Query Processing, G. Bracchi and G.M. Nijessen (ed), Data Base Architecture, IFIP, North Holland, 1979
- S.A. Mahmoud, J.S. Riordon and K.C. Toth, Distributed Database Partitioning and Query Processing Strategies, IFIP Conference on Database Architecture, Venice, June, 1979
- J.S. Riordon, S.A. Mahmoud, K.C. Toth & O. Sherif, Distributed Database Architecture and Query Processing, CIPS/DPMA, Quebec City, June 1979
- K.C. Toth, S.A. Mahmoud, J.S. Riordon, O. Sherif, The ADD System - An Architecture for Distributed Databases, Proc. of the 4th International Conference of Very Large Data Bases, Berlin, September 1978
- S.A. Mahmoud & K.C. Toth, Design Considerations for a Mini-Computer Database, MIMI International Conference, Zurich, June 7-9, 1977
- S.A. Mahmoud, J.S. Riordon & K.C. Toth, Design of a Distributed Database File Manager for a Mini-Computer Network, COMPSAC77, Chicago, November 8-11, 1977
- Kalman C. Toth, Contributions to the Synthesis of Computer-Communication Networks, M.Eng. Thesis, Carleton University, Ottawa, April 1972

## Trade Articles

- "What's the hard part of software development anyway?", Software Assoc. of Oregon, Nov. 2007
- "Better Mileage with Hybrid Learning", with Kathy Milhauser, Software Assoc. of Oregon, June 2007
- "Can Software Engineers Develop Communications Skills Online?", Software Assoc. of Oregon, March 2007
- "Is Online Software Engineering Education for You?", Software Association of Oregon (SAO), Feb 2007
- "OMSE Exchange: A Software Engineering Clearing House", Software Assoc. of Oregon (SAO), Nov 2006
- "So Many Engineering Practices: Which to Follow?" (Part III), Software Assoc. of Oregon (SAO), July 2005
- "So Many Engineering Practices: Which to Follow?" (Part II), Software Assoc. of Oregon (SAO), June 2005
- "So Many Engineering Practices: Which to Follow?" (Part I), Software Assoc. of Oregon (SAO), May 2005
- "Which is the Right Software Process for your Problem?", Software Assoc. of Oregon (SAO), April 2005
- "Outsourcing Software Development: A Case for Effective Scope Management", SAO, March 2005
- "Why Invest in Software Engineering Education?", SAO, February 2005
- "EDI and Security", CGI Group report, Dec. 1990
- "COSICS Security Verification Plan", Intellitech report to External Affairs, December, 1988
- "Information Security Model", report to National Defence, November 15, 1988
- "Data Encryption Equipment Specification", Internal report specifying the components of CryptoNet, Intellitech's X.25/DES product, 1986
- "A Survey of Integrated Project Support Environments", Report to the Department of National Defence, 1986
- "A New Implementation Strategy for Secure Operating Systems", Intellitech Report, March 1986

**Industry Reports**

- "Requirements for SimbaERP", report to Simba Technologies on the Requirements for a proposed ERP/Data Warehousing product, January, 1999
- System Security and Recovery Procedures, Datalink Systems Corp, January 1999
- "Technology Skills Gap Analysis: B.C. Software Industry", under contract to the Software Development Centre (B.C.) for B.C. Ministry of Education, Skills & Training, and National Research Council, March 1997
- "Process Product Standard", internal Hughes System Division Report, June 1994
- "In-Process Review (IPR) Process", internal Hughes System Division Report, December 1993
- "Change in Development Methodology", internal Hughes Systems Division Report, June 1, 1993
- "Total Quality Implementation Program", internal CGI report to the Management Committee, 1991
- "Total Quality Process: Directions & Priorities", internal CGI report to the Management Committee, 1991
- "TQP: Client Satisfaction Assessment Process", internal CGI guide, 1991
- "Software Quality Assurance Program", internal CGI practice guide, 1990
- "Configuration Management Framework", internal CGI practice guide, 1990
- "EDI and Security", CGI Group report, Dec. 1990
- "COSICS Security Verification Plan", Intellitech report to External Affairs, December, 1988
- "Information Security Model", report to National Defence, November 15, 1988
- "Network Processing Strategy Study", a series of reports to Transport Canada, 1988
- "Data Encryption Equipment Specification", Internal report specifying the components of CryptoNet, Intellitech's X.25/DES product, 1986
- "A Survey of Integrated Project Support Environments", Report to the Department of National Defence, 1986
- "A New Implementation Strategy for Secure Operating Systems", Intellitech Report, March 1986
- "Computer System Study" (Computer Integrated Manufacturing and Manufacturing Requirements Planning), Reports to General Metals Co, El Naser Glass Co. and Delta Steel Mills, 1985/86
- "Search and Rescue Satellite (SARSAT) Aided Tracking System, Ground System Study", five reports regarding Mission Control Centre design to National Defence, 1983 and 1984
- "Design Specification for the NCCS Communications Management System", Atmospheric Env. Serv, Jan 84
- "Design & Analysis of Alternatives for the Integrated Data Network", Report to the Dept. Nat'l Defence, 1982
- "Recovery Mechanisms for the ADD Distributed Database System", Intellitech Report, July 1982 (also presented at a NATO workshop in 1982)
- "Implementation Alternatives and Gateway Considerations for a Data Network to Serve the Defence Research Establishments", Report to the Department of National Defence, 1981
- "Design and Security Considerations for a Gateway to Interconnect SAMSON and DATAPAC", Report to the Department of National Defence, 1980
- "Open System Interconnection: Application Issues Associated with the ISO and CCITT Layered Models", report to the Department of Communications, 1980
- "On Query Decomposition & Processing in Distributed DBs", INRIA Research Report, Spyratos & Toth, 1980
- "Query Processing Strategy Formulation in ADD", Carleton University report, 1979
- "A Modeling Approach to Systems Analysis of Processing Networks", one of five reports to the Department of Communications, Spectrum Management Systems
- "Design Issues in Distributed Databases", Carleton University report
- "Design & Configuration Analysis of an Aeronautical Satellite Comm. Centre (ASCC)", Transport Canada



## Workshops, Seminars, Tutorials, Professional Training Courses

- Professional Development Course in Software Engineering for Regence Group, Portland, Or, June 2007
- “Identity Management Systems”, tutorial for IEEE International Computer Software and Applications Conference (COMPSAC), Chicago, September 2006
- Information security seminars for the Assoc. of Prof. Engineers and Geoscientists of B.C., 2002 and 2006
- Extending the Reach of Mobile E-Commerce, Software Productivity Centre, June 2000
- Wireless Handheld Technologies and Telelearning, Telelearning Conference, Toronto, November 2000
- E-Commerce Lifecycle, Transactions and Security, MacDonald Dettwiler & Assoc., November 1999
- Personal Software Process (PSP): Software Productivity Centre / MacDonald Dettwiler & Associates, 1997
- WestMOST Software Engineering Telelearning Workshop, Saskatoon, 1998
- Software Project Management (including software process and metrics) at Carleton University, Dec 1994
- Software Development Methods and Process: Iterative Software Development, for the Canadian Automated Air Traffic System (CAATS) at Hughes Aircraft, Systems Division and Transport Canada, March 1993
- Canadian Automated Air Traffic System, seminars presented at UBC (Computer Science), SFU (Applied Sciences), and Hughes (for staff and graduate students from UVIC, BCIT, SFU and UBC), 1993 and 1994
- Total Quality Management, seminars presented to CGI Group technical staff across Canada, 1991 and 1992
- Total Quality Management, lecture to 4th year computer systems engineers at Carleton University, 1991
- Information Security Technology Overview for AFCEA INFOSEC Course, Canadian Forces Base (CFB) Kingston, October 1991

## University Undergraduate and Graduate Courses

For Portland State University:

- Principles of Software Engineering
- Software Project Management
- Software Quality Engineering
- Software Design Techniques
- Software Estimating
- Distributed Software Engineering Team Collaboration
- Software Engineering Practicum
- Computing Fundamentals II (Visual Basic)
- Senior Capstone projects
- Directed studies: IT and software engineering

For Oregon State University:

- E-Commerce Systems
- Software Engineering I: principles, processes, requirements, OO design, architecture, SPM
- Software Engineering II: implementation, SCM, test techniques, reviews and inspections, SQA

For the Technical University of British Columbia and the University of Alberta:

- Software Engineering Best Practices
- E-Commerce Systems

For the University of British Columbia and Simon Fraser University:

- Software Engineering Best Practices
- Software Project Management
- Professional Issues in Software Engineering
- Software Engineering Team Project

For Carleton University:

- Undergrad course on data structures, databases, programming, and computer architecture